

+ Introduction to DevOps

DevOps is a set of practices that combines software development (Dev) and IT operations (Ops) to improve collaboration, automate processes and accelerate the delivery of high-quality software. By integrating continuous development, testing and deployment, DevOps bridges the gap between development and operations teams, ensuring a seamless workflow.

The primary goal of DevOps is to deliver applications and services at high velocity while maintaining reliability and scalability. Key practices include continuous Integration (CI), Continuous Deployment (CD), Infrastructure as code (IaC), and automated testing. DevOps plays a vital role in modern software engineering, enabling organizations to adapt quickly to changing market demands.

This lab focuses on exploring the essential tools and techniques used in DevOps, such as Git, Jenkins, Docker and Kubernetes, providing hands-on experience to understand the workflow and benefits of this methodology.

1. Perform version control using git/github - install git, create github account, initialize git repository, make commits, creating and merging branches (using CLI/GUI).

Steps to Create Github Account

- 1) Search Github in the browser.
- 2) Open the first link to setup Github account
- 3) Click on Signup
 - Enter mail ID
 - Create Password
 - Create User name
- 4) After sign up, click on Sign In and enter required essential details.

Initializing Git Repository

- 1) Click on Create new repository, select public
- 2) Give repository name, set it to public and initialize repository with a README file
- 3) Click on Create repository

Creating a new file

- 1) In the repository created, click on Add file and select "Create new file".

- 2) Give a name to the file like "test.txt"
- 3) Type something in the text file and then click on commit changes
- 4) Now add a message in the extended description and click on commit
- 5) Now, direct to the repository and add a new text file & in the same way we followed to create first file
- 6) Now, we will create another branch that is "Test" while committing whereas we added the first file to the main branch.
- 7) Click on propose changes, and create a pull request
Now, we can see two branches in our repository.
- 8) Creating a pull request to merge both the branches
- 1) Once you click on propose changes, we will encounter one more page where it shows as "open a pull request"
- 2) Give some description regarding the pull request and then click on create pull request.

Teacher's Signature : _____

3) Now we need to ~~the~~ merge the pull request that was made.

e) Merging pull request

1) Now, we can see the option Merge pull request. click on that

2) once you click that, we need to confirm the merge

3) Then, we can see "pull request successfully merged and closed"

4) when we go to the main branch, we can see the both created files in the same branch.

To perform version control using github integration and collaboration - Push local commits to github and manage pull request, fork a repository make changes to forked repo and merge the changes (using CLI/GUI).

steps

1). Create a new repository

2) ~~Add~~ a new file in the created repository and commit changes

3) Add one more file in the same branch

4) Click on the Fork and create Fork.

~~Adding file in forked repo making local commit in
2nd account~~

5). Click on add file and create 3rd file & commit changes

6) After adding 3rd file, click on contribute → click "open pull request"

7). Once the "open pull request" opens, add the description and then click on create pull request

8). The owner of the repo will accept the pull request and merge the changes.

Teacher's Signature : _____

9) we can see the pull request in the main account

10) Accept the request, Click on "Merge pull request" →
click "confirm merge"

11) Once the merge is complete, changes will be visible
in main -repo.

Using CLI

>> git clone https://github.com/cdc/program.git

>> cd program

>> touch test.txt

>> git add .

>> git commit -m "Initial Commit".

>> git push

>> git branch b1

>> git checkout b1

>> touch test2.txt

>> git add .

>> git commit -m "branch commit".

>> git push

Then rest of the process is same as we followed for GUI.

Creation of YAML files to determine define workflow configuration in github repository for CI/CD pipelining.

1) Create a simple python file in any IDE

addition.py

```
def add(a,b):  
    return(a+b)
```

```
def test_add():  
    assert add(1,2)==3  
    assert add(2,2)==4
```

2). Adding a yml file to the .github/workflows directory is a part of setting up Github Actions

3) The yml defines the steps for automating specific tasks such as testing, building or deploying your code

4) once the yml file is created, we need to push the code into github repo.

5) click open "git bash" in the folder created

6) Go to settings in github → Actions → Runners → create new runner → select runner & execute commands in git bash

Teacher's Signature : _____

7) Go to actions in github → view workflow file → we can view the execution of the pipeline

Config

name: first GitHub Actions #Name of the GitHub Actions workflow

on: [push] # Trigger this workflow on push events.

jobs:

build: # Job name

runs-on: ubuntu-latest # The job runs on the latest version of ubuntu

strategy:

matrix:

python-version: [3.8, 3.9]

steps:

step 1: check out the repository code into the runner.

- uses: actions/checkout@v2

step 2: set up the specified python versions from the matrix

- name: Install dependencies

- name: Setting up Python

uses: actions/setup-python@v2

with:

python-version: \${{ matrix.python-version }}

Teacher's Signature : _____

step-3: Install all the ~~dependencies~~ required for the tests

- name: Install all the ~~dependencies~~ dependencies
reen: 1

python -m pip install --upgrade pip # upgrade pip
pip install pytest.

step-4: Run the tests using pytest

- name: Running tests

reen: 1

python -m pytest addition.py

Commands to execute in git bash +

\$ mkdir actions-runner

\$ cd actions-runner

curl -L -o actions-runner-win-x64-2.319.1.zip https://github.com/actions/runner/releases/download/v2.319.1/actions-runner-win-x64-2.319.1.zip

unzip actions-runner-win-x64-2.319.1.zip -d .

./config.cmd --wd <github repo> --token <token number>

./reen.cmd

Teacher's Signature : _____